Drawing lattices with a geometric heuristic

Jan Outrata

Abstract—**Lattices play an important role in many areas of computer science and applied mathematics. The information, extracted from data in data analysis or operated with in intelligent systems, is usually represented by hierarchical structures, where relationships are described by lattices. To visualize the information, one needs to visualize lattices. We mention the existing methods of automated drawing of lattices and focus on the geometric method introduced by Wille et al. Diagrams drawn by the geometric method achieve a good level of readability and aesthetic criteria while satisfying common conventions and constraints, even for larger lattices. We discuss several questions regarding the method and show the diagram drawings produced by two software programs developed at Dept. Computer Science, Palacky University, Czech Republic.**

Index Terms— **lattice drawing, automated drawing, geometric heuristic, Hasse diagram**

I. Introduction

A. Motivation

Many new disciplines of computer science and applied mathematics, from data analysis and information retrieval to machine learning and intelligent systems, present information in the form of hierarchically ordered structures. From the theoretical point of view, the hierarchy structures are usually described by lattices. Hence, with the increasing interest in the disciplines using lattices in recent years, one needs to visualize lattices more often than before.

Lattices, as a class of ordered sets, are visualized best by a good and easily readable drawing of the Hasse diagram. The Hasse diagram of a lattice is a graph with certain constrains and conventions used for two-dimensional drawing of the diagram. The definition of a Hasse diagram states the requirement saying that the drawing of a node representing the lesser element of the lattice should be placed bellow the drawing of a node representing the greater element. Hasse diagrams are also sometimes called upward or linear diagrams, meaning that the drawing of an edge is a straight line segment without arrows. In the rest we will not distinguish the diagram and the diagram drawing, without the danger of confusion.

Jan Outrata is with Dept. Computer Science, Palacky University, Olomouc Tomkova 40, CZ-779 00 Olomouc, Czech Republic, e-mail: jan.outrata@upol.cz

The research was supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

The quality, the high level of readability, of the diagram, which is required by a user, is determined by the actual layout of the drawing, i.e. the arrangement of nodes and lines on the plane. However, the requirement from the definition of Hasse diagram and other drawing conventions say nothing about the diagram layout. Human aesthetics criteria assuring a good level of readability and used for drawing the diagram by hand include optimization requirements like minimizing the number of line crossings (optimally planar drawing with no line crossings), minimizing the number of slopes, maximizing the conflict distance (i.e. the least distance of a node from a non-incident line), maximizing the angle between incident lines, maximizing the number of symmetries by arranging nodes in an orthogonal grid and others [6], [13], and the empirical analyzes about their importance highlight the first criterion. Despite that, nobody knows exactly what makes the best readable diagram.

Drawing diagrams by hand was quickly automatized by computer programs and several different algorithms and methods for drawing (layouting or arranging) diagrams based on heuristic approaches were developed. Unfortunately, the automatic arrangement of diagrams remains a difficult task, since it is by no means obvious how to mathematize the (yet no clear) human aesthetics criteria. Nevertheless, although drawings of the diagram of a lattice by hand are traditionally better, they are tedious to create compared to the automatically generated ones, especially for lattices counting tens or more of elements in which case the drawing by hand is almost impossible. Furthermore, the automatically generated drawings serve as a good starting point for final fine-tuning by hand.

B. Existing approaches and methods for drawing lattices

Much work has been done in drawing graphs [6] and many approaches to drawing lattices borrow thoughts and techniques developed in graph drawing. Among the most commonly used are the layered approach [6] and the force directed approaches [9]. The basic characteristic of the layered approach is that the nodes are drawn on the imaginary horizontal lines (the layers) based on their distance from the top or bottom node (i.e. the number of nodes on the minimal path between the node and the top or bottom node), depending on whether we draw the diagram top-down or bottom-up direction. Hence the nodes are drawn layer by layer. Nodes in the same layer are sorted using the optimiza-

978-1-4244-1739-1/08/$25.00 © 2008 IEEE

tion techniques to achieve the minimal number of line crossings. The approach is generally accepted in drawing lattices too, though it does not much respect the symmetries criterion, for instance. The force directed approaches are based on imaginary repulsive and attractive forces between nodes and lines with the aim to relax the nodes and put the forces in balance. This approach demonstrates the natural way to optimize the conflict distance criterion, on the account of the other criteria, however.

The drawing methods using these and other approaches for drawing graphs were more or less successfully adapted for Hasse diagrams of lattices [3], [4], [5], [7], [22]. Also, new interesting approaches suited for lattices have been developed. For instance the quite popular approach of attribute additivity [4], [?]. In this approach the position of a diagram node on the plane is given by the sum of vectors assigned to greater or equal infimum-irreducible elements (or dually less or equal supremum-irreducible elements) of the lattice. The approach provides the overall geometric regularity of the diagram with many parallel lines, fulfilling the symmetries criterion. Besides the methods adapting graph approaches, there also have been developed quite a few special methods for drawing lattices [12], [15], [16], [17], [19], [20], especially due to the need for visualizing concept lattices used in Formal Concept Analysis (FCA) [10].

All existing methods for drawing lattices need no initial drawing of a lattice (sketched by hand), the input for the methods is the underlying lattice order relation only. Today, the methods produce quite readable diagrams under the consideration of the many aesthetics criteria mentioned above. This is true especially for smaller lattices, counting about twenty or so elements at maximum. However, in the case of larger lattices, counting tens of elements or more, there is certainly a room for improvements. This is because most of the methods use the approach of global optimization of aesthetic criteria for the whole lattice, which is limited by the size of the lattice. In the paper we are interested in the geometric method which uses the local optimization approach, which provides quite readable diagram drawings even for larger lattices.

II. The geometric heuristic

A. Geometric method

We first recall the geometric method for drawing lattices as was for the first time proposed in [16] and [19] and re-introduced by Wille et al. in [17]. The method is one of the special methods originally developed for drawing concept lattices in FCA. The geometric method is based on a geometric interpretation of the lattice and viewing the lattice theoretical structure through the geometric representation called a geometrical diagram. The geometrical diagram is an auxiliary picture describing the lattice order relation in a graphical fashion which resembles the view on a three-dimensional visualization of the lattice from its top element. For geometrical diagram illustrations and details on drawing the diagram we refer to [17].

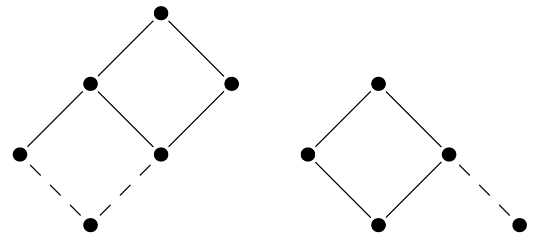

Fig. 1. Rule of parallelograms (left) and rule of lines (right)

The key part of the method is then finding the best possible layout of the Hasse diagram of the lattice using the geometrical diagram. The process of drawing the Hasse diagram consists in recognizing certain geometric patterns in parts of the diagram drawn and realizing the patterns. In the process mainly two following geometric rules are used:

Rule 1 (Rule of parallelograms) A new node should be placed in such a way that the node together with some already placed nodes and lines forms a parallelogram (the geometric shape with parallel lines, e.g. diamond or rhomboid).

Rule 2 (Rule of lines) A new node should be placed on a prolonged line connecting some already placed nodes, preferably at the same distance as the distance between the nodes.

Altough the rules are looking simple general rules, there are several questions in applying the rules. The immediate problem is the vague formulation of "some already placed nodes". The most commonly selected nodes are (1) in the case of the rule of parallelograms the nodes of a pair of upper (when drawing the diagram top-down) neighbor lattice elements together with the node of their common upper neighbor element (lattice supremum), see Fig. 1 (left, suggested node placement displayed by dashed lines); (2) in the case of the rule of lines the nodes of the single upper neighbor element and an upper neighbor of that neighbor, see Fig. 1 (right).

The application of the rules results in many parallel lines and regular geometric shapes (diamonds and rhomboids) in the diagram, fulfilling the symmetries aesthetics criterion and all other mentioned criteria as well. The application of the rules is the essential part of the general process of discovering regular geometric shapes, structures and patterns in the diagram drawing, aiming of the best possible overall geometric regularity of the drawing. Fig. 2 shows some illustrative examples of Hasse diagrams drawn by the geometric method and Fig. 3 depicts the diagrams of all 10-element lattices with four infimum-irreducible elements [1].

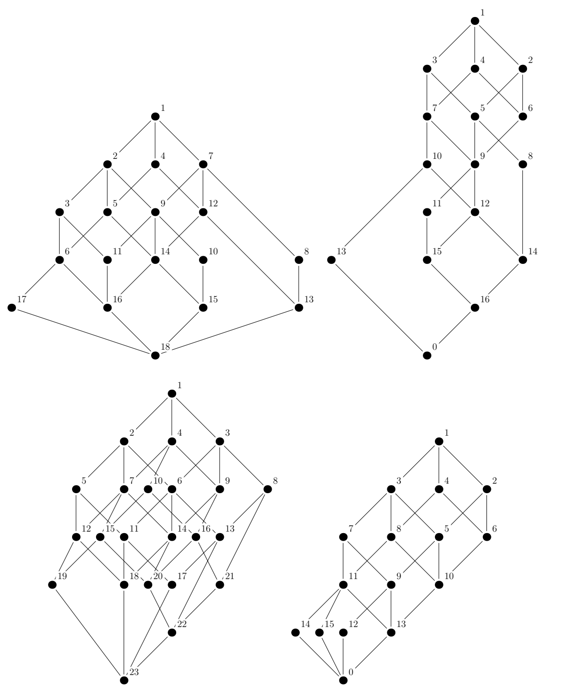

Fig. 2. Illustrative examples of the geometric method

B. Open questions

However, the application of the rules is not a straightforward action at all, there are many decision points. Looking back at the rule of parallelograms, one can see that there can be more than one pair of upper neighbor elements. Which one to choose to form a parallelogram with the actual newly arranged node? This situation is illustrated in Fig. 4. It seems the preferred pair of elements should be the one for which their supremum is as “close” as possible to them (ideally the upper neighbor), forming as small and easily readable parallelogram as possible (the one connected with the bold face lines in Fig. 4). Still there can be more possibilities. Or if the supremum is not an upper neighbor, should we place the newly arranged node in the middle below the nodes of all upper neighbor elements, forgiving the rule of parallelograms? Analogously, in the case of the rule of lines there can be more than one upper neighbor of the single upper neighbor element, as illustrated in Fig. 5. Here the preferred one is not so easy to see. Or should we place the newly arranged node straight bellow the node of the single upper neighbor element? Usually the final choice depends on suggested arrangements of nodes of further (lower) elements.

Furthermore, we have to arrange the first (top or bottom) nodes before we can apply the rules at all. When drawing the diagram top-down those are the nodes of lower neighbors of the top element (i.e. the co-atoms of the lattice). Similarly this applies to the nodes of lower neighbors of an element with the single upper neighbor, for which the rule of lines suggests the same location. We can place them on an imaginary horizontal line or on a parabola, or we can take more

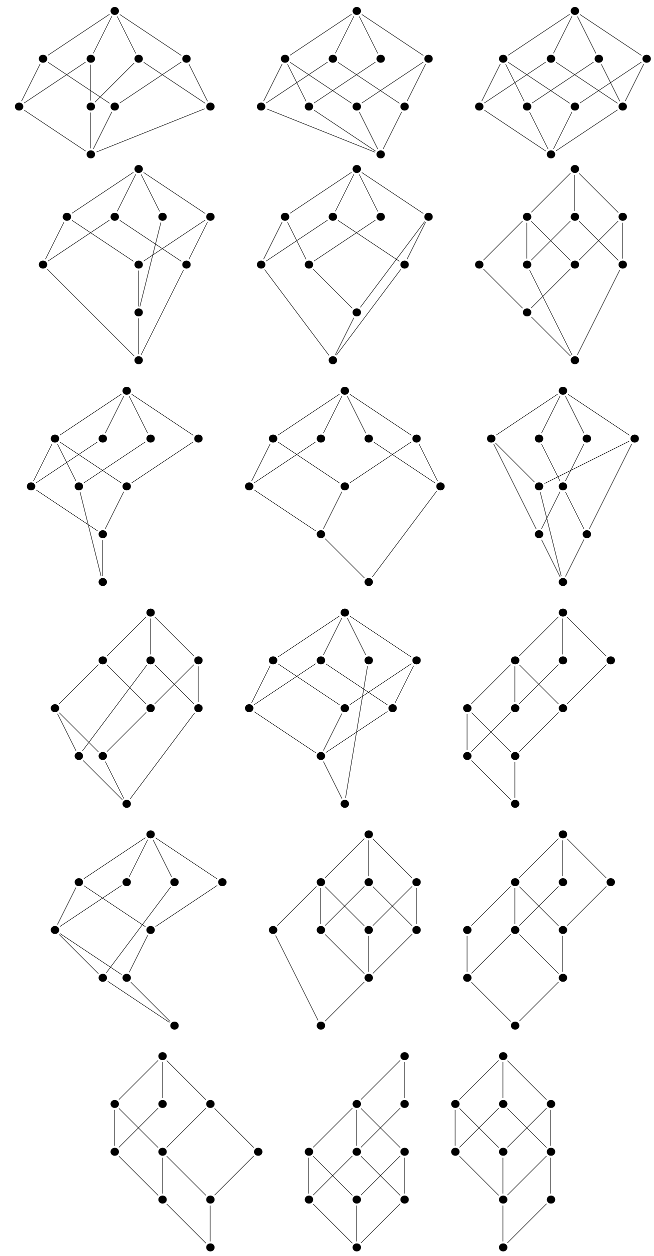

Fig. 3. 10-element lattices with four infimum-irreducible elements

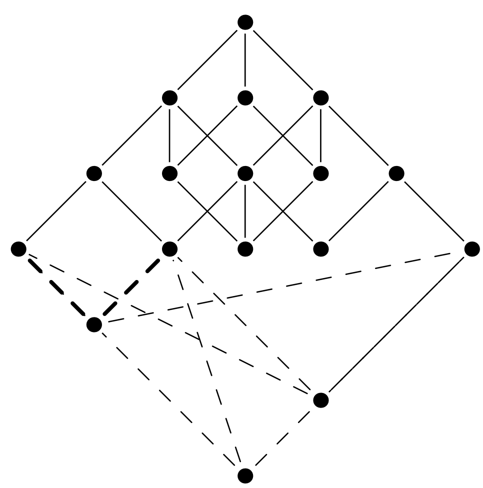

Fig. 4. Placements suggested by the rule of parallelograms

[1] The lattices were obtained by a tool described in [2]

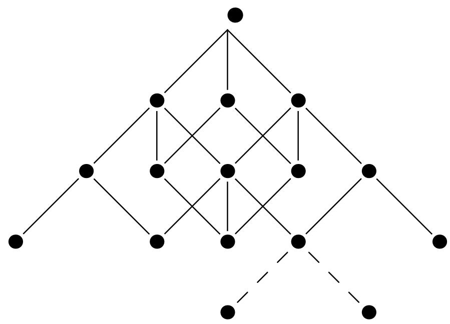

Fig. 5. Placements suggested by the rule of lines

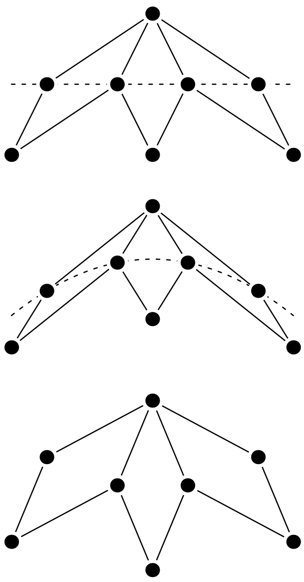

Fig. 6. Different arrangements of co-atoms

sophisticated placement using the force directed approach, see Fig. 6. In either case the additional problem of the order of the elements is arising. Here, the elements having more lower neighbors should be placed aside with the elements having less lower neighbors in the middle. This is advocated by the need of space for the lower elements of those elements. The same principle can be used for arranging nodes with the same best evaluated location suggested by the rules or other way.

Besides the questions and problems mentioned above, there arise many other "little nifty" decision problems to solve in the process of drawing the diagram by the geometric method. The whole process of layouting the diagram thus consists in the sequence of several nontrivial heuristic solutions to local optimization problems of optimal diagram node arrangement.

III. Software for drawing lattices

We know, every method of automated drawing of lattices has to be implemented in software. First to test and evaluate the method, then to further develop and tune it to finally make it best usable by end users. Such a software should, besides producing the best possible drawing of the lattice from the lattice order relation only, have a features like subsequent fine-tuning the produced diagram layout by hand (or by additional final touching and layout enhancing methods if those are not a part of the layout generation method already), manipulating parts of the diagram (moving and aligning nodes, folding, fine-tuning etc.), exporting the final diagram or parts of it to the picture easily usable in a paper and so on.

Some of the existing methods mentioned in section I-B are implemented in software programs [1], [8], [18], [21], but in general there is very few software for this task and usable by end user at the same time. Needless to say, the original purpose of the most of the available programs is a tool for FCA and drawing concept lattices is one (but indeed important) of the features only. Moreover, the elements of a concept lattice have a special structure which is often further utilized by the drawing method used. The main disadvantage of such specialized programs thus is that they may be a bit complicated to use for drawing of arbitrary (not concept) lattices, or even ordered sets. The features of subsequent fine-tuning the diagram or the parts of it by hand and exporting to a picture easily usable in a paper are of limited extent, too.

This is where our programs, LatVis (2003) [14] and EllenaArt (2007) [11], take place. The programs were developed as an accompanying software of MSc. diploma theses of the author and his student at Dept. Computer Science, Palacky University, Czech Republic and are depicted in Fig. 7 (LatVis) and Fig. 8 (EllenaArt). Both programs were originally designed for drawing arbitrary lattices (or even ordered sets in the case of LatVis) and have the feature of fine-tuning the produced diagram by hand, aligning the nodes to a grid structure (EllenaArt) and exporting the final diagram to the picture of METAPOST or Encapsulated Postscript format (the former can also be compiled into the PDF format).

The programs contain preliminary implementations of the original geometric method. The implementations are almost without any heuristics and the problems mentioned in the paper are solved in an arbitrary manner (usually the first possible solution, i.e. the first suggested node placement, is used). All examples of diagrams in the paper were produced the two programs.

In addition, both programs contain (naive) heuristic methods developed by authors and inspired by the layered approach, LatVis contains an implementation of a generic graph method exploiting the layered approach and adapted to lattices and EllenaArt contains implementations of the couple of three adapted graph methods exploiting the force directed approach. However, from the comparison of the implemented methods, performed in the theses, the geometric method came out in both cases with the best drawn lattices as a winner. This was actually the first impulse for us to decide to further develop and refine the geometric method by using sophisticated heuristics, to give even more readable Hasse diagrams of lattices.

IV. Conclusion and forthcoming research

Interestingly, after the proposal papers [16], [17], [19] there are virtually no further papers on the geometric method for drawing lattices. We find the method very promising regarding the level of readability and aesthetic criteria of the resulting diagram layout. Therefore, in the present paper we have discussed several problems encountered in the realization of the method and proposed some ideas to tackle the problems. The paper shall be followed by other papers describing our method of automatic drawing of lattices, inspired by and further refining the geometric method.

The main idea of our method is using an intermediate description of the resulting diagram layout similar to the geometrical diagram used in the original geometric method. However, the description, preliminary called the logical diagram description, is more general and used thoroughly the process of arranging the resulting diagram layout. The description contains information like the constrains of Hasse diagram regarding the arrangement of nodes w.r.t the layout of nodes of lower and upper neighbor elements, space constraints for arranging lower or upper neighbor elements (which can influence the order of processing of elements), as well as possible placements suggested by the two geometric rules and other principles, evaluations of the placements (which is used to choose the final placement) etc. The resulting Hasse diagram is then obtained from the logical diagram description by a heuristic process of solving both local and global optimization problems of determining the best final layout of the resulting diagram.

References

[1] P. Becker, J. Hereth, G. Stumme: ToscanaJ – an open source tool for qualitative data analysis. FCAKDD 2002, pp. 1-2, 2002, `http://toscanaj.sourceforge.net/`.

[2] R. Belohlavek, V. Vychodil: Scales behind computational intelligence: exploring properties of finite lattices. *Foundations of Computational Intelligence*, 2007. FOCI 2007. IEEE Symposium on 1–5 April 2007, pp. 556–563.

[3] R. Cole: Automatic Layout of Concept Lattices using Force Directed Placement and Genetic Algorithms. In: Edwards J. (Ed.): 23th Australiasian Computer Science Conference, pp. 47–53, Australian Computer Science Communications 1, IEEE Computer Society, 2000.

[4] R. Cole: Automated Layout of Concept Lattices Using Layered Diagrams and Additive Diagrams. In Oushoorn M. (Ed.): 24th Australiasian Computer Science Conference **23**, pp. 47–53, *IEEE Computer Society*, 2001.

[5] R.–J. Cole, J. Ducrou, P. Eklund: Automated Layout of Small Lattices Using Layer Diagrams. Proc. of ICFCA'06, *LNAI* **3874**, pp. 291–305, 2006.

[6] G. Di Battista, P. Eades, R. Tamassia, I.–G. Tollis: *Graph Drawing: Algorithms for the Visualisation of Graphs*, Prentice-Hall, New Jersey, 1999.

[7] R. Freese: Automated lattice drawing, In: Proc. 2nd Int. Conference on Formal Concept Analysis, pp. 112–127, *LNAI* **2961**, Springer, 2004.

[8] R. Freese: Online Java Lattice Building Application. `http://www.math.hawaii.edu/~ralph/LatDraw/`

[9] T. Fruchterman, E. Reingold: Graph Drawing by Force Directed Placement. *Software – Practice and Experience* **21**, no. 11, pp. 1129–1164, 1991.

[10] B. Ganter, R. Wille: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.

[11] L. Hostalek: EllenaArt, `http://phoenix.inf.upol.cz/~hostalel/en/ellenart/ellenart.html`

[12] P. Luksch, M. Skorksy, R. Wille: On drawing concept lattices with a computer. In Gaul W., Schader M. (Eds.): Classifcation as a tool of research, pp. 269–274, North Holland, Amsterdam, 1986.

[13] M. Mller-Hannemann: Drawing Trees, Series-Parallel Digraphs, and Lattices. In: M. Kaufmann and D. Waggner (Eds.): Drawing Graphs, LNCS 2025, pp. 46-70, 2001, Springer-Verlag Berlin Heidelberg.

[14] J. Outrata: LatVis, `http://phoenix.inf.upol.cz/~outrata/latvis/`

[15] M. Skorsky: *Endliche Verbände–Diagramme und Eigenschaften.* Ph.D. thesis, TH Darmstadt, Shaker Verlag, 1992.

[16] M. Skorsky: How to draw a concept lattice with parallelograms. In Wille R. (Ed.): Klassifikation und Ordnung, pp. 191–196, Indeks-Verlag, Frankfurt, 1989.

[17] G. Stumme, R. Wille: A geometric heuristic for drawing concept lattices. In: Tamassia R., Tollis I. G. (Eds.): Graph drawing, pp. 85–98, Springer, Berlin-Heidelberg-New York, 1993.

[18] P. Valtchev, D. Grosser, C. Roume, M.–R. Hacene: GALICIA: an open platform for lattices. In: Using Conceptual Structures, Contributions to the 11th Intl. Conf. on Conceptual Structures (ICCS'03), pp. 241–254, Dresde, Shaker Verlag, 2003, `http://www.iro.umontreal.ca/~galicia/`.

[19] R. Wille: Geometric representation of concept lattices. In: Opitz O. (Ed.): Conceptual and numerical analysis of data, pp. 239–255, Springer-Verlag, Berlin-Heidelberg, 1989.

[20] R. Wille: Lattices in data analysis. How to draw them with a computer. In: Rival I. (Ed.): Algorithms and order, pp. 33–58, Kluwer, Dordrecht-Boston, 1989.

[21] S. Yewtushenko et al.: ConExp, `http://conexp.sourceforge.net/`

[22] Ch. Zschalig: An FDP-Algorithm for Drawing Lattices. In: Diatta J., Eklund P., Liquire M. (Eds.): Proc. CLA 2007, pp. 58–71, LIRMM & University of Montpellier II, 2007.

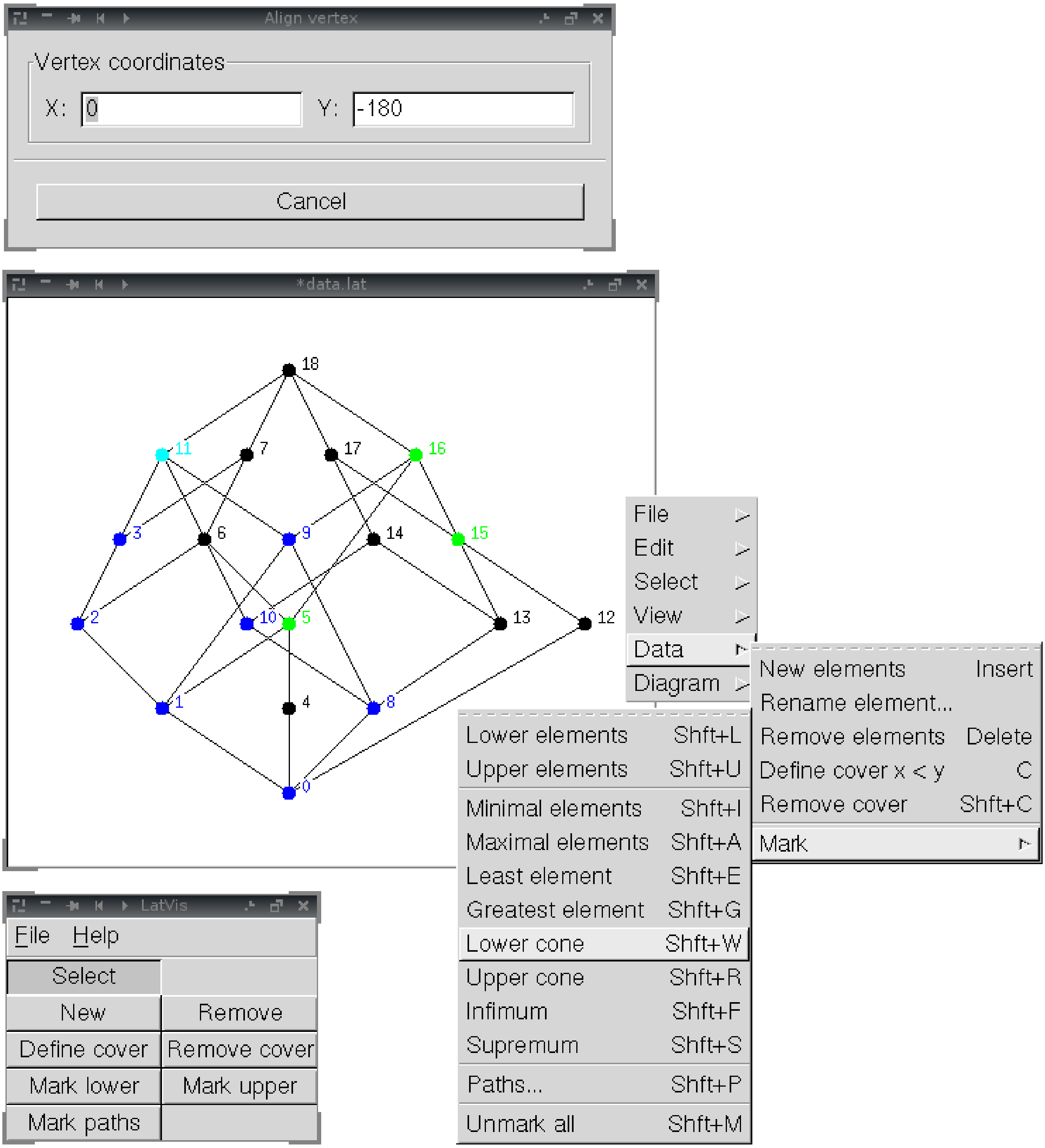

Fig. 7. LatVis (screenshot)

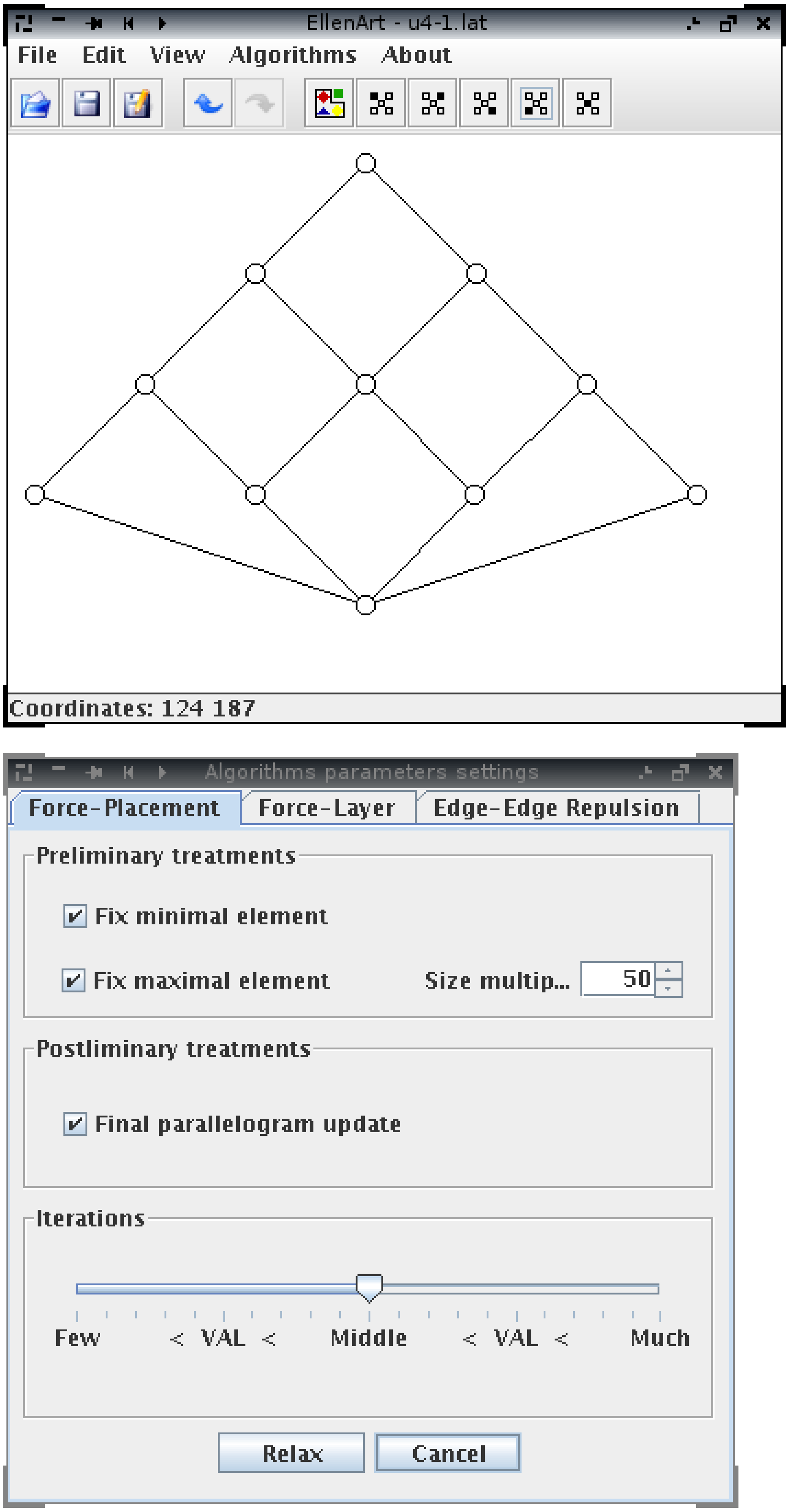

Fig. 8. EllenaArt (screenshot)